**PÁGINA WEB – RESTAURANTE CINCO HOJAS**

****

AUTOR: VÍCTOR SÁENZ MARTÍNEZ

TUTOR: RAFAEL ALAMAÑAC

PROYECTO FIN DE CICLO – DAW

CURSO 2022-2023

# **Índice**

**Contenido**

[**Índice** 1](#_Toc135591461)

[**1 Introducción** 1](#_Toc135591462)

[**2 Planificación** 1](#_Toc135591463)

[**3 Análisis** 3](#_Toc135591464)

[**3a Alcance** 3](#_Toc135591465)

[**3b Objetivos** 4](#_Toc135591466)

[**3c Requisitos funcionales y no funcionales** 4](#_Toc135591467)

[**4 Diseño** 5](#_Toc135591468)

[**4a Diagrama de casos de uso** 5](#_Toc135591469)

[**4b Configuración de la base de datos** 6](#_Toc135591470)

[**4c Diagrama de base de datos:** 7](#_Toc135591471)

[**4d Diseño de clases** 7](#_Toc135591472)

[**5 Desarrollo e Implementación** 9](#_Toc135591473)

[**6 Pruebas** 25](#_Toc135591474)

[**7 Ampliación y posibles mejoras** 27](#_Toc135591475)

[**8 Conclusión** 28](#_Toc135591476)

[**9 Bibliografía** 28](#_Toc135591477)

# **1 Introducción**

Mi proyecto consiste en una página web para una empresa ficticia, un restaurante. Esto le permitirá tener una identidad en la web y ser reconocido por mas gente.

Su principal objetivo es promocionar el restaurante en un nuevo canal, ofreciendo la carta, que consiste en una serie de menús degustación, y permitir a los clientes reservar una mesa en el local.

La idea al elegir esta temática surgió al intentar encontrar información sobre un restaurante en internet y comprobar que este no tenía página web. Sin duda tener un espacio en la web es indispensable hoy en día para la promoción de un negocio.

Esta aplicación pretende mostrar los productos del negocio de forma accesible y clara, pudiendo explorar los platos que ofrece cada menú con sus ingredientes principales y la información sobre alérgenos. También es posible reservar en el restaurante a través de esta aplicación web.

# **2 Planificación**

Para agilizar la planificación a la hora de desarrollar la aplicación he utilizado un modelo de metodología ágil, SCRUM. Esto facilita la tarea dividir el proyecto en fases o sprints y ponerme plazos para la realización de los mismos. Cada fase será planificada con anterioridad, se evaluará el resultado que se quiere obtener en cada una y un tiempo estimado para su realización. Asimismo, al finalizar el desarrollo de una fase se hará una pequeña prueba para encontrar posibles fallos o mejoras.

**Fases en la creación de la aplicación:**

**1 Creación de la interfaz de la página principal:**

* Plazo de 4 horas para su realización

**2 Desarrollo del header, nav y footer**

* Plazo de 2 horas para su realización

**3 Desarrollo y conexión de la BD (MySql):**

* Plazo de 3 horas para su realización

**4 Creación del login y registro de usuarios:**

* Desarrollo de la interfaz y el código necesario para registrar usuarios en la BD
* Plazo de 5 horas para su realización

**5 Creación del diseño de la página “Carta”:**

* Plazo de 2 horas para su realización

**6 Desarrollo de la funcionalidad de la carta:**

* Plazo de 2 horas para su realización

**7 Creación del diseño de la página “Platos”:**

* Plazo de 3 horas para su realización

**8 Desarrollo de la funcionalidad de la página “Platos”:**

* Plazo de 2 horas para su realización

**9 Creación del diseño de la página “reserva”:**

* Plazo de 3 horas para su realización

**10 Desarrollo de la funcionalidad de la reserva:**

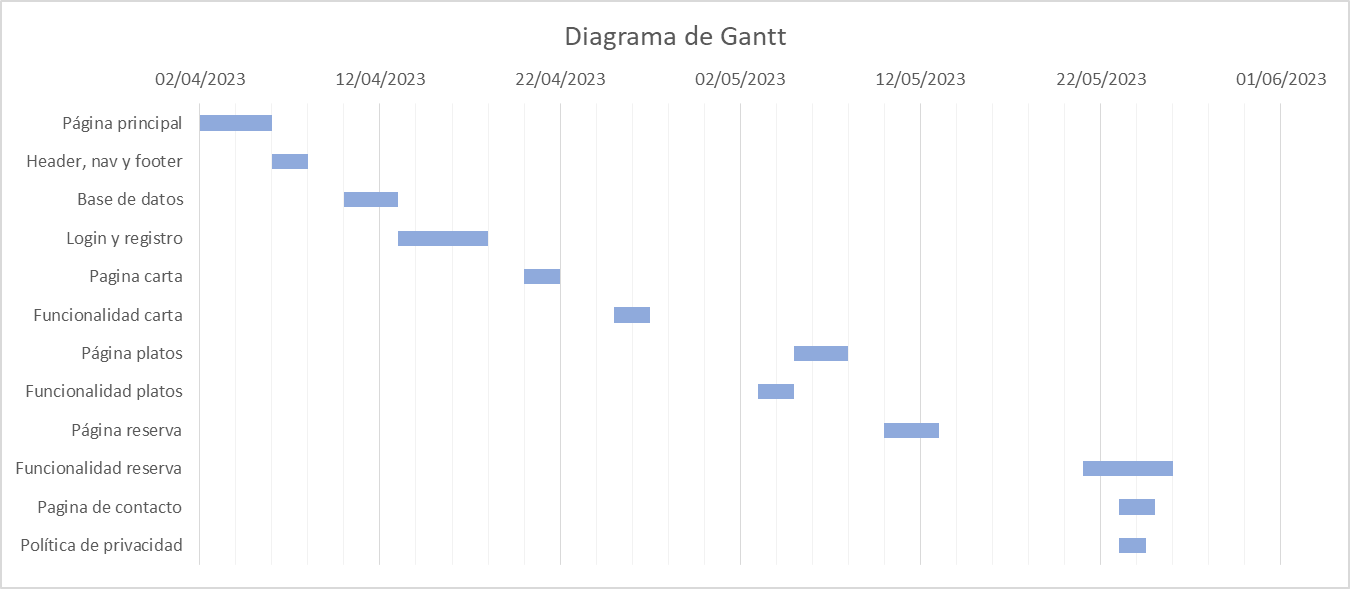
* Plazo de 5 horas para su realización

**11 Creación del diseño de la página de contacto “conócenos”:**

* Plazo de 2 horas para su realización

**12 Creación de diseño de la página “Política de privacidad”:**

* Plazo de 1.5 horas para su realización



# **3 Análisis**

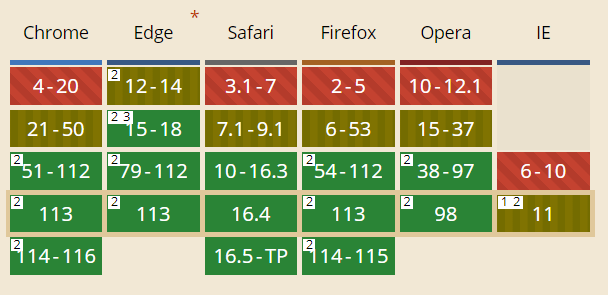
## **3a Alcance**

Dentro del alcance de la aplicación vamos a definir para que usuarios o dispositivos está orientada la app:

**En cuanto al dispositivo para el que está desarrollada:**

Al ser una aplicación web, estará disponible para cualquier navegador ya sea en un teléfono móvil, una tablet o un ordenador.  
Respecto a la versión de las tecnologías utilizadas, aclarar que PHP es interpretado y ejecutado por parte del servidor y no depende de nuestro navegador.

El uso de la versión de JavaScript “ES6” sí que podría poner en peligro el buen funcionamiento de la aplicación en algunas versiones de los navegadores. Esto podría ocurrir en la versión de Chrome 20 y anteriores, la de Firefox 5 y anteriores, etc. Adjunto una imagen con la información más detallada:



La aplicación solo va a estar disponible en castellano, lo que limita a los usuarios que no hablen este idioma.

**En lo respectivo a la funcionalidad:**

Al tratarse de una página informativa, esta aplicación puede ser usada por cualquier persona. La opción de reserva de mesa en el local tampoco supone una limitación a la edad de los usuarios.

## **3b Objetivos**

El objetivo principal es darle una identidad a la empresa en la web, darse a conocer en este ámbito. La página mostrara todos los productos de dicha empresa, permitirá al público informarse de que tipo de cocina ofrece este restaurante.

Dado que los medios informáticos están siendo la plataforma más usada a la hora de contratar un servicio o adquirir un producto, debido en gran parte a la comodidad de la gestión, muchos usuarios pueden estar interesados en reservar una mesa en el restaurante a través de la aplicación web.

El sistema de cuentas de usuario facilita en gran medida el trámite al cliente, ya que después de registrarse en la página, podrá reservar con solo acceder y elegir el dia y la hora. Este registro solo será obligatorio a la hora de reservar, la información de los menús será accesible a cualquier usuario que visite la página.

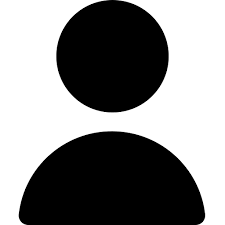
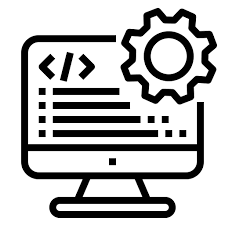
## **3c Requisitos funcionales y no funcionales**

|  |  |
| --- | --- |
| REQUISITOS FUNCIONALES | |
| RF-1 | La aplicación es multiplataforma, para móvil, PC y Tablet. |
| RF-2 | Permite hacer reservas en el restaurante |
| RF-3 | El usuario deberá registrarse para reservar |
| RF-4 | Se podrán crear cuentas de administrador para editar las reservas |

|  |  |
| --- | --- |
| REQUISITOS NO FUNCIONALES | |
| RNF-1 | Interfaz simple y llamativa |
| RNF-2 | Muestra todos los menús y los platos del restaurante |
| RNF-3 | Muestra toda la información de contacto del restaurante |
| RNF-4 | Deberá dejar identificarse a todos los usuarios |
| RNF-5 | Muestra los ingredientes y alérgenos de cada plato |
| RNF-6 | Muestra la imagen de cada plato |

# **4 Diseño**

## **4a Diagrama de casos de uso**



Usuario

Aplicación

Valida y almacena las credenciales

Introduce credenciales

Inicio Sesión/

Registro

Envía los datos del menú correspondiente al usuario

Elegir menú en la página “carta”

Consultar Menús

Vuelca los datos de las reservas en un calendario

Acceder a la página reserva y consultar el calendario

Consultar la disponibilidad

Elegir el dia y la hora de la reserva, y el número de comensales

Reservar Mesa

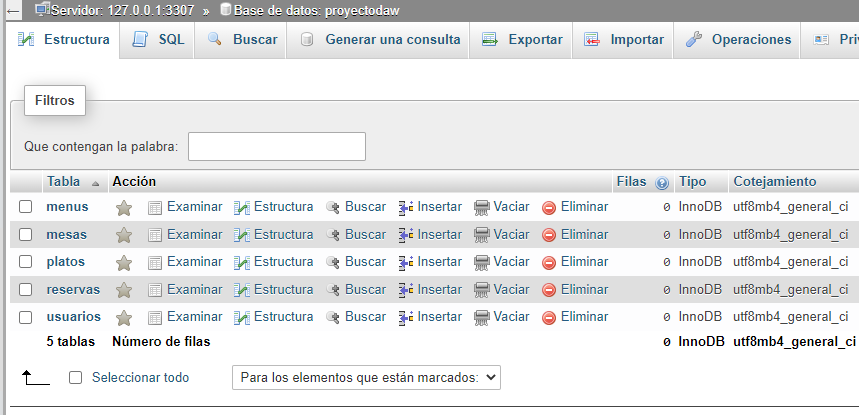
Añade un registro a la tabla reservas y bloquea la mesa establecida a esa hora

## **4b Configuración de la base de datos**

Para el almacenamiento y la gestión de los datos de la aplicación he usado la base de datos MySql, que tiene una estructura relacional. La información se dispone en tablas cuya estructura es definida por sus columnas.

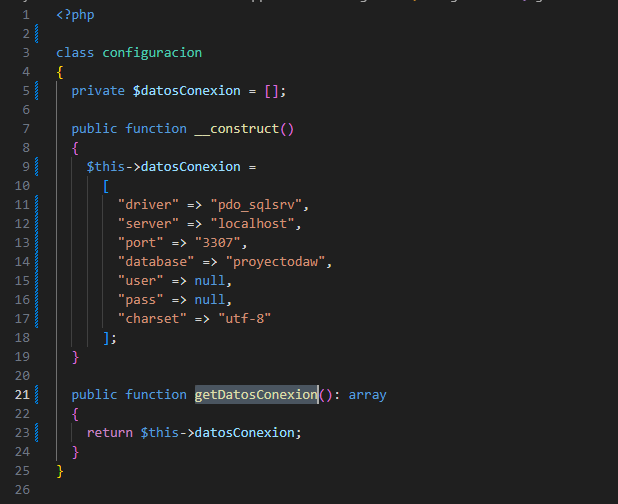
La base de datos estará albergada en Xampp, que facilita la gestión y conexión con el servidor Mysql a través de la aplicación PHPmyAdmin.

Una vez configurada la conexión, creamos la base de datos con el editor que nos ofrece la aplicación.



Para abarcar todos los datos del proyecto han sido necesarias cinco tablas, una que representa a los usuarios, y servirá para almacenar sus credenciales y sus datos personales. En otra tabla se almacenarán todos los platos, estos serán referenciados por la tabla “menus” a través de la clave foránea “IdPlato”. Para persistir los datos de cada reserva se añadirán las claves foráneas de las tablas “usuarios” y “menus” a la tabla “reservas”.

Para conectar la base de datos con la aplicación he creado un archivo PHP que almacenará los datos de la conexión y otro que la creará, servirán de respaldo para todas las clases que requieran acceder a la base de datos.



Clase donde se definen los datos para la conexión



Clase donde se crea la conexión

## **4c Diagrama de base de datos:**

IdReserva

Cancelado

Hora

Fecha

Comensales

IdMesa

DniUsuario

**Reservas**

**Menús**

IdMenu

Precio

IdPlato1

IdPlato2

…

IdPlato13

1 - N

**Platos**

IdPlato

TipoPlato

Alérgenos

Nombre

Descripción

Imagen

N

**Mesas**

IdMesa

NumMesa

Capacidad

1 - N

1 - N

**Usuarios**

Dni

Nombre

Apellidos

FechaNacimiento

Correo

Contraseña

EsAdmin

1 - N

1

* Un plato puede estar en más de un menú.
* Un menú tiene más de un plato.
* Un usuario puede tener una o más reservas.
* Cada reserva está hecha por un usuario.
* A cada reserva se le asigna una mesa.

## **4d Diseño de clases**

|  |
| --- |
| EventoCalendario |
| + IdReserva: Numérico  + DaniUsuario: Cadena  + IdMesa: Numérico  + Comensales: Numérico  + Fecha: Date  + Hora: Cadena  + Cancelado: Bool |

|  |
| --- |
| Usuario |
| + Dni: Cadena  + Nombre: Cadena  + Apellidos: Cadena  + Fecha Nacimiento: Date  + Correo: Cadena  + Contraseña: Cadena  + EsAdmin: Bool |

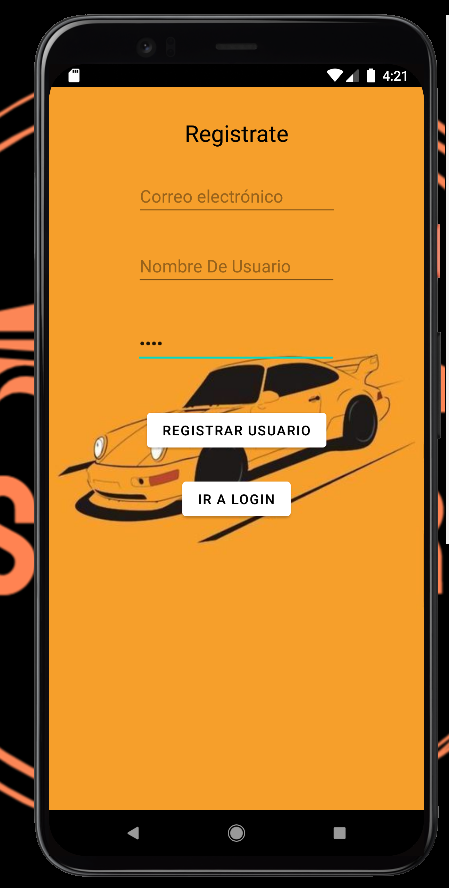
|  |
| --- |
| Plato |
| + IdPlato: Numérico  + TipoPlato: Cadena  + Alergenos: Cadena  + Nombre: Cadena  + Descripción: Cadena  + Imagen: Cadena |

# **5 Desarrollo e Implementación**

**Pantalla de Registro**

Su función es recoger datos de los nuevos usuarios para que estos puedan empezar a usar la aplicación.

**Interfaz:**



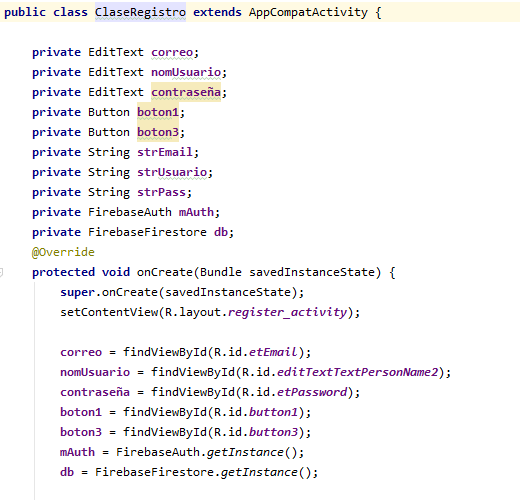
**Funcionamiento:**

Uno de los botones te redirecciona al login, en caso de que ya tengas cuenta y quieras iniciar sesión.

Al introducir los datos en los editText y pulsar el botón “registrar usuario”, añade las credenciales a la base de datos, y abre la pantalla principal.

Para gestionar el correo y la contraseña, valores que usaremos cada vez que queramos iniciar sesión en la appc función de Firebase dedicada a este fin: FireBaseAuthentication. Esta clase se comunica con un gestor de credenciales de la base de datos, proporcionando una forma segura de acceder a tu cuenta y simplificando en gran medida el código necesario para registrarse e iniciar sesión.

Creamos la instancia a la clase FirebaseAuth.



llamamos a su método createUserWithEmailAndPassword, y le pasamos el valor de los campos de texto de “correo” y “contraseña”.

El propio metodo tiene por defecto una función de prevención de errores al estilo Try-Catch.

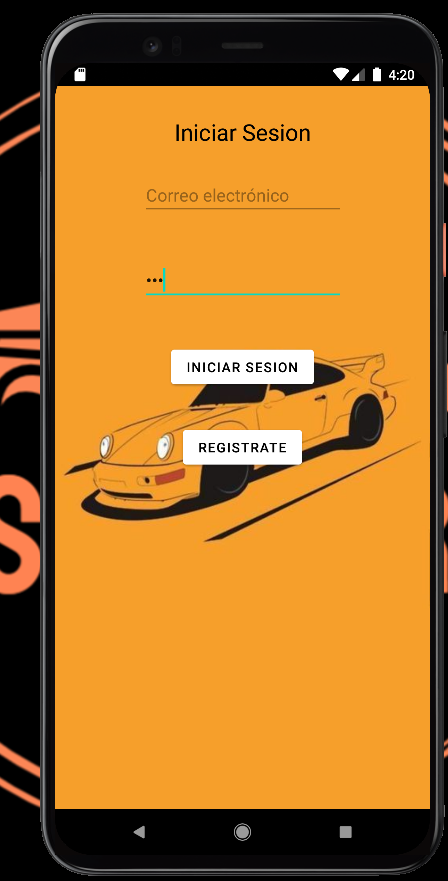
Si la transacción sale bien, almacena los datos del usuario en una colección de la base de datos, y en el gestor de credenciales de Firebase.



**Pantalla de login**

Permite al usuario iniciar sesión con sus credenciales, o ir al registro para crear una cuenta.

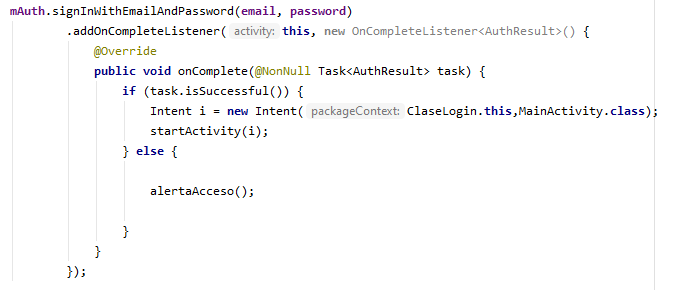
**Interfaz:**



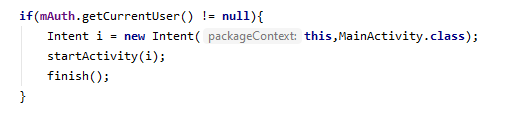
**Funcionamiento:**

Al introducir las credenciales correctamente y pulsar en iniciar sesión, accede a la pantalla main de la aplicación con el usuario especificado.

Al igual que en el registro, las credenciales son administradas por Firebase authentication. En este caso se usa el método “singnInWithEmailAndPassword”, al que se le administran las credenciales introucidas, y él se encarga de gestionar la validación.



Firebase también proporciona una función que permite recordar el último usuario que inició sesión en la aplicación y saltarse el login, ya que valida las credenciales automáticamente. Está gestionado por Firebase Authentication.



**Pantalla principal**

Pantalla principal de la aplicación, muestra un resumen de los datos del alquiler del coche, en caso de que el usuario esté alquilando alguno. En caso contrario, invita a reservar uno. En la parte de abajo tiene un menú de navegación que permite desplazarse entre las pantallas de la aplicación. Hay uno similar en cada pantalla.

**Interfaz:**





**Funcionamiento:**

El cartel de información está cargado con los datos de la colección ‘cochesPrestados’, que almacena los datos referentes a los préstamos. Se envía una consulta a dicha colección, restringiendo por nombre de usuario. Si el usuario no tiene datos en cochesPrestados, significa que no tiene ningún coche en alquiler. Entonces se muestran los datos de la segunda captura.

El menú de navegación está formado por tres botones enlazados con cada una de las pantallas, con un icono representativo.

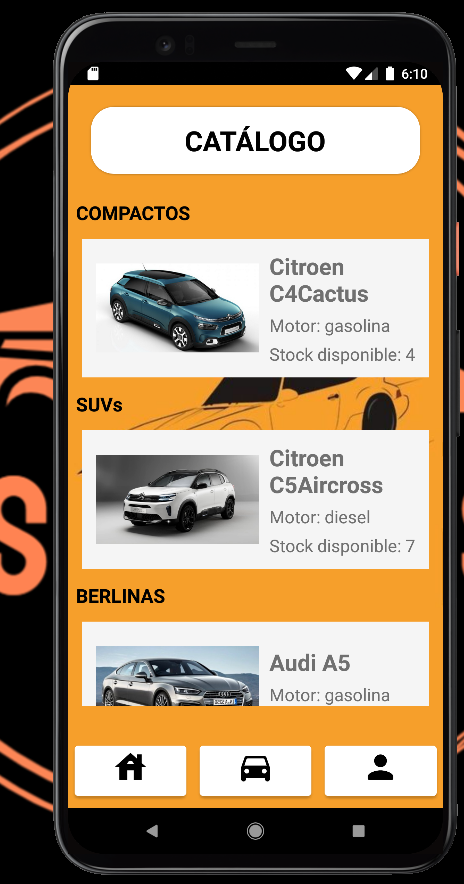
También dispone de un botón que permite cerrar sesión con el usuario activo y volver al login.

**Catálogo**

Es la interfaz donde se encuentran todos los coches disponibles para alquilar, volcados desde la base de datos. Al Pulsar en el elemento de cualquier coche se muestra un diálogo con la información del mismo, y un pequeño formulario para su alquiler.

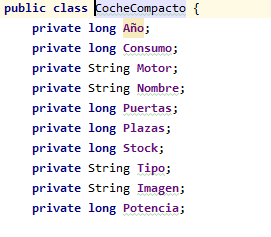
**Interfaz:**

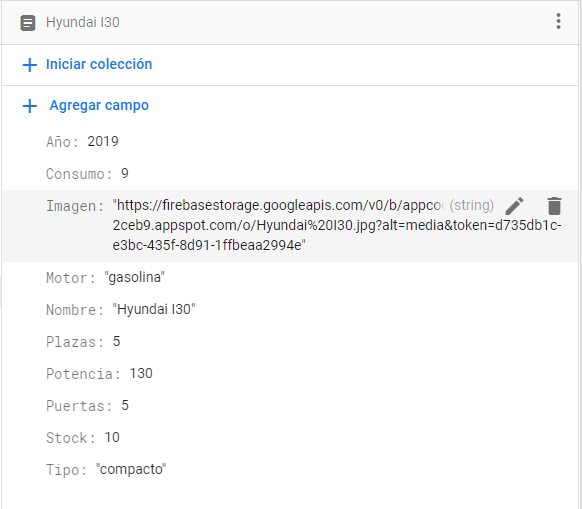
Cada categoría de coches se muestra en un recycler view horizontalmente, los ítems del recycler view llaman a un diálogo, interfaz que explicaré en el siguiente apartado.



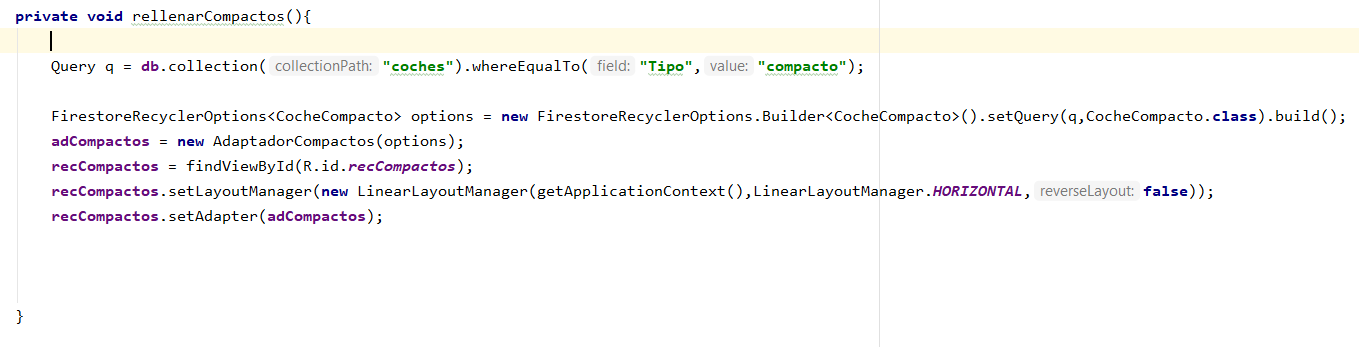
**Funcionamiento:**

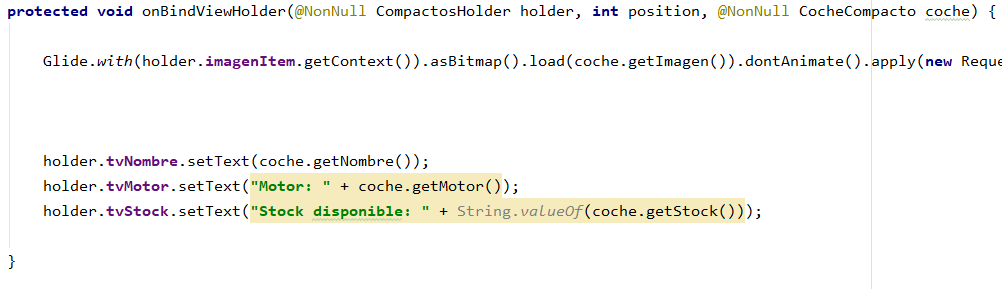
Los datos de cada categoría se muestran en un recycler view. Firebase aporta un modo de implementación para los recyclerViews, simplificando la carga de datos desde la colección determinada. Cada documento que queremos recuperar de la bd se mapea en un objeto de java, previamente construido con los campos del documento que nos interesa obtener.





Una vez ejecutada la consulta, se genera un objeto con los valores de los campos del documento, que son recuperados a través de los getters del mismo.





**Diálogo Alquiler**

Esta interfaz se muestra al pulsar en un ítem de los recyclerView del catálogo, muestra toda la información del coche elegido, y permite su reserva, eligiendo la fecha en la que se debe devolver el coche.

**Interfaz:**

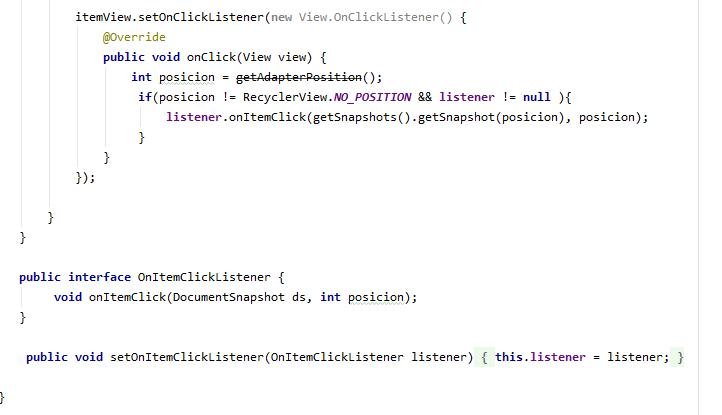


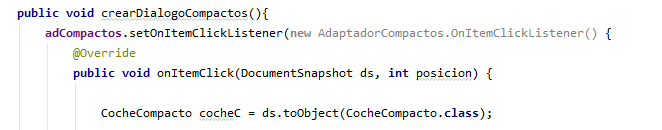
**Funcionamiento:**

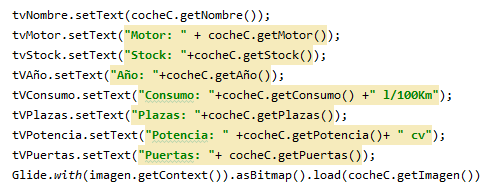
El diálogo se genera a partir de los datos del RecyclerViewItem pulsado, ya que la captura del evento está en el adaptador de dicho reciclerView.

Este método listener enviará al dialogo abierto una instancia de la clase CocheCompato en este caso, al pertenecer al tipo “compactos”. Esta transferencia se realiza gracias a un DocumentSnapshot.

Con este objeto recibido se obtienen todas las propiedades del coche a partir de sus getters.







La imagen, como expliqué anteriormente, se genera a partir de la url almacenada en el campo de la base de datos, utilizando la librería “Glide”

Esta interfaz cuenta con un selector de fecha, que muestra días desde la fecha actual. Con este control el usuario decide cuando tiempo va a tener el coche en alquiler, ya que es la fecha de devolución del coche.

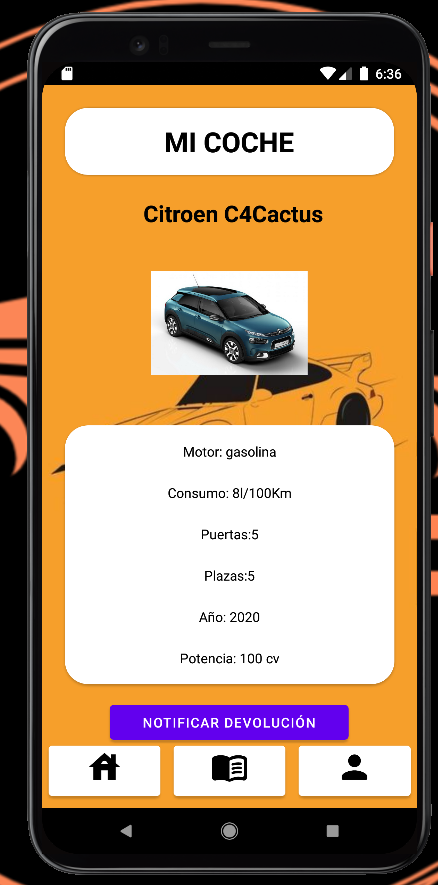
Para acabar, si el usuario decide reservar el coche, hará click en el botón reservar coche, lo que creará un nuevo documento en la colección cochesPrestados de la bd, con los datos del alquiler.

**Mi Coche**

Interfaz en la que se visualizarán los datos relacionados con el coche alquilado por el usuario, en caso de tenerlo.

**Interfaz:**





**Funcionamiento:**

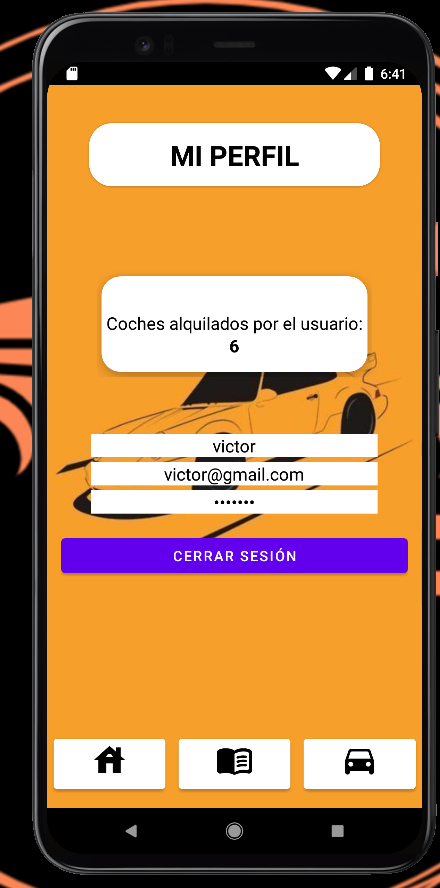
En caso de no tener ningún coche en alquiler, se mostrará la interfaz de la primera captura, al alquilar un coche, se cargan los datos del mismo desde el documento “Coches”, filtrando por el campo común entre las colecciones coches y cochesPrestados: nomCoche.

Esta interfaz permite notificar la devolución del coche a la empresa, esta acción desactivará el alquiler del coche por parte del usuario.

**Mi perfil**

En esta pantalla se mostrarán los datos del usuario activo, y un registro de los coches que ha alquilado en total.

**Interfaz:**



**Funcionamiento:**

El usuario activo se identifica con la función FirebaseAuthentication, que recupera los datos de la sesión actual. El número de coches alquilados totales es un valor acumulativo cargado desde la base de datos

La interfaz también proporciona la opción de cerrar la sesión.

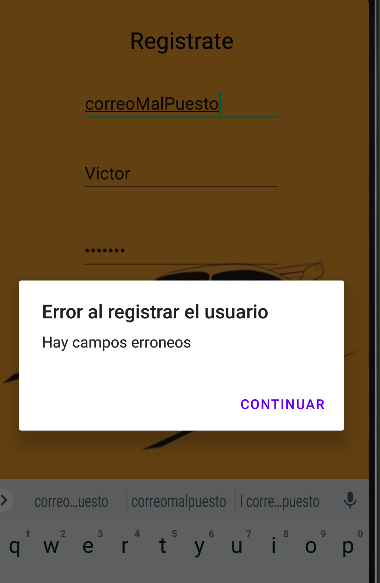
# **6 Pruebas**

En esta sección explicare las comprobaciones que he ido realizando durante y después de la realización del desarrollo de la aplicación.

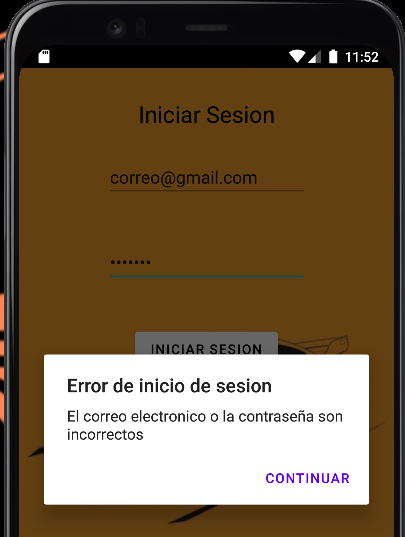
Para estas pruebas, he utilizado el dispositivo virtual de Android Studio. He elegido un dispositivo móvil virtual de tamaño medio (5,7’), que representa la experiencia media que tendría un usuario utilizando la aplicación en su dispositivo.

En primer lugar comprobamos que el registro no permita introducir campos erróneos, especialmente en el EditText que corresponde al correo electrónico.

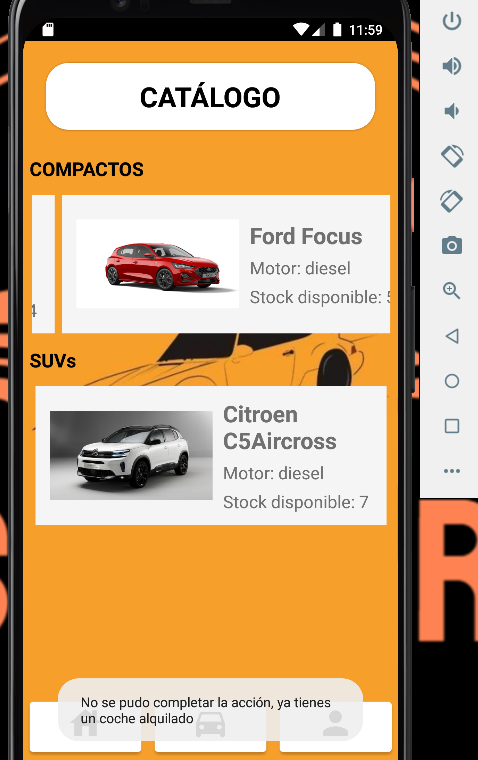
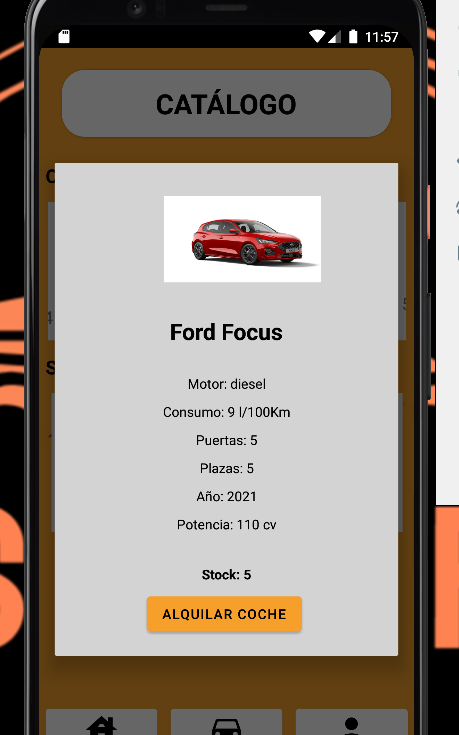
Vemos que si no se introduce el formato de correo adecuado, el programa no permite registrar al usuario.



Siguiendo el recorrido que realizaría el usuario, comprobaremos el comportamiento del login, cuando no se introducen las credenciales correctamente.



Ahora intentaremos alquilar un coche teniendo ya un alquiler activo, el programa no lo permite

# **7 Ampliación y posibles mejoras**

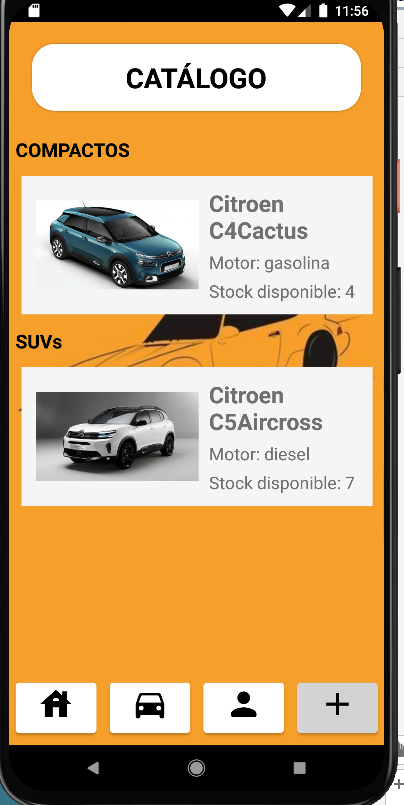
La aplicación cuenta con las funciones básicas necesarias para el alquiler de un coche y el volcado de datos de los diferentes apartados de una empresa de este tipo, pero hay una gran cantidad de funcionalidades que se podrían implementar y mejorarían la experiencia del usuario a la hora de usar esta app en su teléfono.

En esta sección hare un resumen de las que me resultan más viables.

**Creación de usuarios administradores:** a la hora de expandir el catálogo añadiendo nuevos automóviles a la base de datos, la única opción es hacerlo directamente desde la base de datos. Añadiendo esta función, un representante de la supuesta empresa podría añadir coches fácilmente desde un formulario solo visible para este tipo de usuarios.

Darse de alta como administrador requeriría una clave especial, que al introducirla en el registro daría la opción al usuario de ser administrador de la aplicación.

Este formulario se podría asociar a un botón en la barra de navegación de la pantalla “catálogo”, solo visible para administradores.



**Notificación de devolución:** la aplicación podría lanzar una notificación al usuario cuando queden pocos días para la devolución del coche en alquiler.

**Información sobre el combustible:** el combustible consumido en la utilización de un coche de alquiler se tiene que reponer antes de devolverlo. Por eso una pestaña informativa sobre el consumo y precio de este en diferentes estaciones de servicio sería de gran utilidad para el usuario.

Por una parte, el precio del combustible se obtendría de los datos en abierto que publican ciertos organismos de forma informativa, sería un valor a tiempo real.

Por otra parte se implementaría una calculadora del consumo, eligiendo el precio al que se vende el combustible en la gasolinera elegida por el usuario, introduciendo los kilómetros recorridos, se generaría el coste del recorrido realizado.   
Esta función puede ser útil a la hora de organizar un viaje, pues haría parte del trabajo a la hora de calcular el coste total.

wireframe etc, capturas de la pagina

admin: cancelar reservas, festivos

# **8 Conclusión**

Esta aplicación puede resultar muy útil a empresas que trabajan el sector del alquiler de coches, y quieren dar a conocer su negocio en las redes.

Tener disponible en el móvil, una herramienta que todos nosotros usamos diariamente, la posibilidad de informarse sobre los coches de los que dispone la empresa, y la posibilidad de reservar uno desde cualquier lugar, darían gran ventaja a la empresa en cuanto a visibilidad y marketing digital.

Con las mejoras que he expuesto en el apartado de ampliación, la aplicación sería mas completa y sería de utilidad incluso a la hora de planificar viajes y negocios, aumentando así los usuarios potenciales.

# **9 Bibliografía**

* <https://es.stackoverflow.com/>
* <https://caniuse.com/>
* https://www.w3schools.com/